Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2015), *29*, 483–502.
© Cambridge University Press 2015 0890-0604/15
doi:10.1017/S0890060415000463

Notes on the design process of a responsive sun-shading system: A case study of designer and user explorations supported by computational tools

RODRIGO VELASCO,[1,2] RUBÉN HERNÁNDEZ,[3] NICOLÁS MARRUGO,[3] AND CÉSAR DÍAZ[4]

[1]Programme of Architecture, Universidad Piloto de Colombia, Bogotá, Colombia
[2]Façade Engineering, Frontis3D, Bogotá, Colombia
[3]Programme of Mechatronics Engineering, Universidad Piloto de Colombia, Bogotá, Colombia
[4]Design Department, Frontis3D, Bogotá, Colombia

(RECEIVED September 15, 2014; ACCEPTED June 23, 2015)

Abstract

Responding to growing concerns regarding energy-efficient facades, this paper describes the structure and process followed in the design of a responsive sun-shading system based on the use of rotating plates with two degrees of freedom. The proposal considers, among others, the definition of variable design parameters, areas of performance evaluation and control, and construction detailing development represented by a first 1:2 unit (module) model. In the process, computational simulation procedures were employed to explore configurational possibilities that would provide high-performance solutions to the light requirements of the particular covered spaces. In developing the system, it was noticed that due to the highly subjective requirements of users in terms of quantity and quality of lighting, a purely Boolean control system would not always be appropriate. Following from that, and taking advantage of the dynamic nature of the system, a further approach of control supported by fuzzy logic was also implemented at the operative state, whose logic is explained. Digital simulations were carried out to assess the performance of the system, and their results demonstrate more even light distribution levels compared to traditional systems.

Keywords: Architectural Lighting; Control Systems; Dynamic Facades; Fuzzy Logic; Interactive Design

1. INTRODUCTION

The advent of widely available computational tools to plan, simulate, and control dynamic processes, in conjunction with the introduction and lowering of prices of related electronic equipment including different kinds of microprocessors, sensors, and actuators, has meant a significant growth not only in research but also in the actual implementation of dynamic responsive facades during the last few years. At the same time, taking into account the increasing need for energy-efficient buildings to cope with the current environmental crisis, dynamic facade systems seem to offer new grounds for hitherto unattainable high-performance solutions. The system described here is inscribed in the area of dynamic facades and focuses on its functional behavior as a light filter. However, the discussion of the design structure and its process will explain particular ways in which the implementation of Boolean and fuzzy logic control systems have been used to support design explorations at the design and operational stages.

1.1. Dynamic facades

A dynamic facade is one that changes properties to procure variable configurations through a period of time. In this broad view, the change of its components may not necessarily be related to physical movement, because it is the case of changing light production or transmission properties found in many dynamic facade examples, the first case normally procured by means of interactive LED components and the second usually using suspended particle devices, liquid crystal, or electrochromic layers embedded within laminate glass elements.

This paper limits the scope of dynamic facades to those whose changing properties are attributable to physical motion of the material elements constituting the system. In that respect, and emphasizing the control component by means of computation, there are various recent research projects with specific denominations for the dynamic system, among others: Climate Adaptive Building Shells (Loomen, 2010) Bioclimatic Responsive

Reprint requests to: Rodrigo Velasco, Programme of Architecture, Universidad Piloto de Colombia, Carrera 9 #45, Bogotá, Colombia. E-mail: rodrigo-velasco@unipiloto.edu.co

Skins (Urquiza, 2010) Adaptive Solar Envelope (ASE; Rossi et al., 2012) Intelligent Building Skins (El Sheik, 2011), Responsive Envelopes (Thün & Velikov, 2013), and Advection Based Adaptive Building Envelopes (Vollen & Winn, 2013).

The broad scope of sometimes overlapping terms and the sharp increase in literature including forthcoming texts on the subject suggests that this area of research is still in development, and following that, the generic term *dynamic facades* will be used, even if narrowed by the clarifications made above. In order to contextualize this project in the general area of dynamic facades, however, two main areas for categorization have been used, the first one describing the actual movement of the facade components, and the second one describing the control structure that drives its behavior. In this context, a new inclusive classification based on movement (Stevenson, 2011) and control (Fox & Kemp, 2009) has been proposed (Velasco et al., 2015), according to which the present system would be described as a second-degree off-plane rotation mechanical movement, operated by a central, system-based control structure.

1.2. Problem definition

When talking of daylight requirements for an architectural space, there are at least two main topics that have to be taken into account, on the one side, the amount of light that enters the space, which is normally measured in percentage as daylight factor (but preferably measured as an absolute magnitude for a specific site and orientation in lux), and on the other side, the actual distribution of that light inside the space. The latter topic is particularly important, taking into account that an average value, even if very high, may imply direct daylight (above 20,000 lux, depending on location) near windows, but at the same time extremely low lighting levels in the interior of the space (below 100 lux), which would involve important problems of glare. In that respect, recent research suggests that when designing the solar protection for a particular space, visual issues should be separated from what had normally been understood as solar protection (Dubois, 2013). One may take the typical example of solar protection configurations for high latitude sites, where solar rays are normally blocked during summer to avoid overheating, but let through during the winter period where the acquired energy would help reduce the heating loads of the space. This situation, even if it makes sense from a temperature point of view, may imply important problems of glare during the winter period. As for the levels of illuminance, there are ranges that are normally acceptable for human activities (varying between 300 and 2000 lux), and accordingly, there are standards to assess the proportion of the covered space where such levels are available. However, things get more complex because specific activities and human predilections call for differing specific lighting levels. The problem that a dynamic sun-shading system faces is then twofold and relative: how can it provide the specifically required lighting levels inside a space (relative) and at the same time procure them as evenly as possible within the area of such space?

1.3. General configuration of the system

To comply with the requirements previously discussed, that is, avoiding glare and securing (as much as possible) uniform lighting levels inside the covered space, a facade system would need both to filter direct solar radiation and also to redirect solar rays to deeper interior parts of the space. This double function requires not only different spatial orientations of the panels but also a different type of material configurations. While redirecting solar rays demands the use of reflective surfaces, filtering direct radiation may allow for the use of such radiation in the form of energy via photovoltaic (PV) cells. However, because a compromise between reflecting and absorbing capabilities within a single surface would decrease its performance by at least 50%, the system has been designed to make use of the two sides available for each panel, giving each side a specific surface and function, the exterior (PV) exclusively for absorption and the interior (reflective) for light harvesting purposes (Fig. 1).

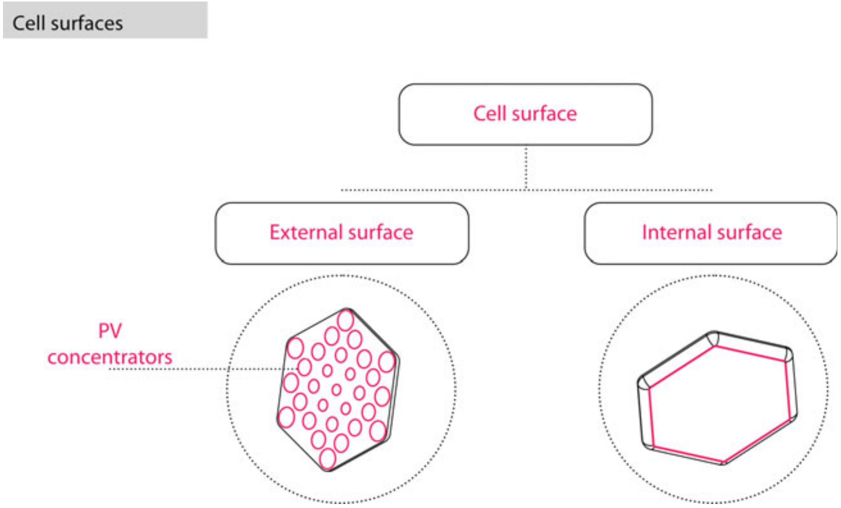

Fig. 1. Proposed two-sided panel.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

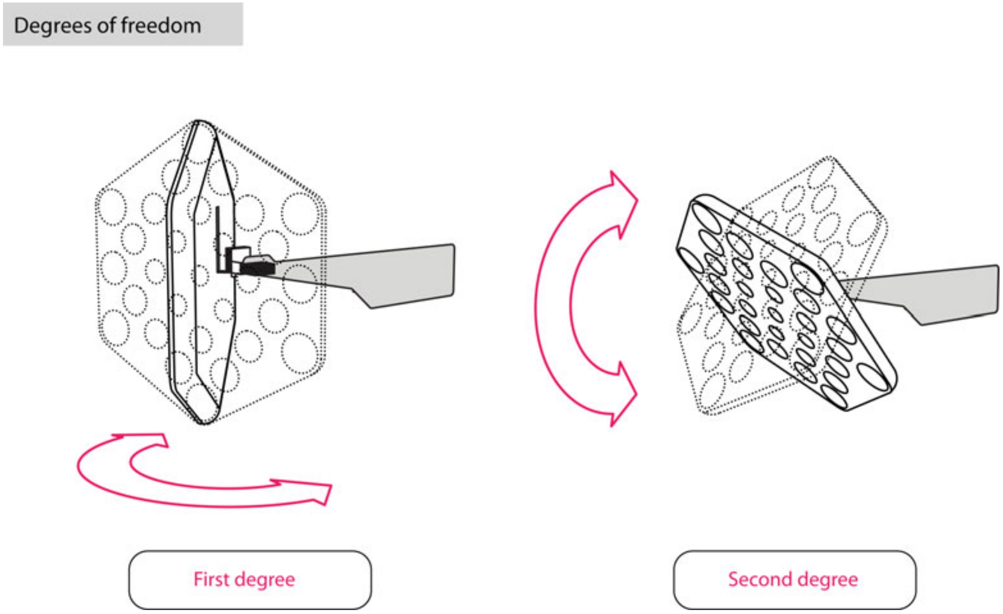

Fig. 2. First and second degrees of movement: hourly and daily rotation directions.

Following from that, the proposed physical configuration is as follows: a facade made out of plates with two functional sides, one with PV energy concentrators and the other exhibiting a highly reflective surface. This would be the fixed definition of the system regarding its physical configuration, but grid geometries and frequencies would be variable parameters (Fig. 2).

In terms of control, the logic is also relatively simple: if more light is required inside the covered space, the panels would reflect solar rays to the interior of the space by exposing the reflective side oriented in space as to reflect the solar rays toward selected interior areas, whereas if the opposite is true, the panels would track the sun exposing the PV side. The implementation, however, is slightly more complex, because various factors are involved in the process; and it is not always straightforward to define when more or less light is required in the interior space.

2. GENERAL LOGIC OF THE SYSTEM

At the most general level, the logic of dynamic facades can be described in the same way as any interactive computing system, that is, by an iterative cycle implying the use of inputs, data processing, and outputs (Moloney, 2007 p. 9). In a more specific way, and common to other mechatronic appliances, the input component would include data taken from the environment and users, either preset or via sensoric devices; the control component would understand the input data as variables and process them following a preestablished logic to produce a data output that would then inform the actions of actuator devices, normally motors, whose actions would in their way have implications on the input data, thus closing the feedback iterative cycle. The logic of this system is thus divided into three components: input, control, and output. The feedback loop that makes the system closed happens here because the resulting movement of the system components (component panels) has consequences on the interior lighting levels measured by interior sensors, which are subsequently used as inputs.

Accordingly, it is clear that the facade system should be able to integrate the particular positions that would allow them to operate as sun blockers or reflectors, given a particular relationship of available light and user settings in the functional behavior of each component. In this context, the use of two axes rotating planar components, including the use of local (component-based) sensors and actuators within a central control system, already implies that the main outcome of our logic should be the two values for angle definition that will place the component plates in position. However, the way in which inputs are processed to define such angles and the actual computational and material implementation of the system should still be clarified (Fig. 3).

2.1. Input variables

To organize the input variables, four general types have been defined: *design data, predictable environmental conditions data, sensor gathered data*, and *user defined data*, this last one being particularly important because it marks the relativeness of the system and allows user overriding. Each one of these types contains different kinds of input data as follows.

2.1.1. Design data

This involves the geometric definitions of the facade, its context and components, being the main area for the design explorations at a first stage, because this data would be processed along with the other inputs to render different outputs that are evaluated in terms of performance according to preset lighting targets. These variables are the building geometry

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

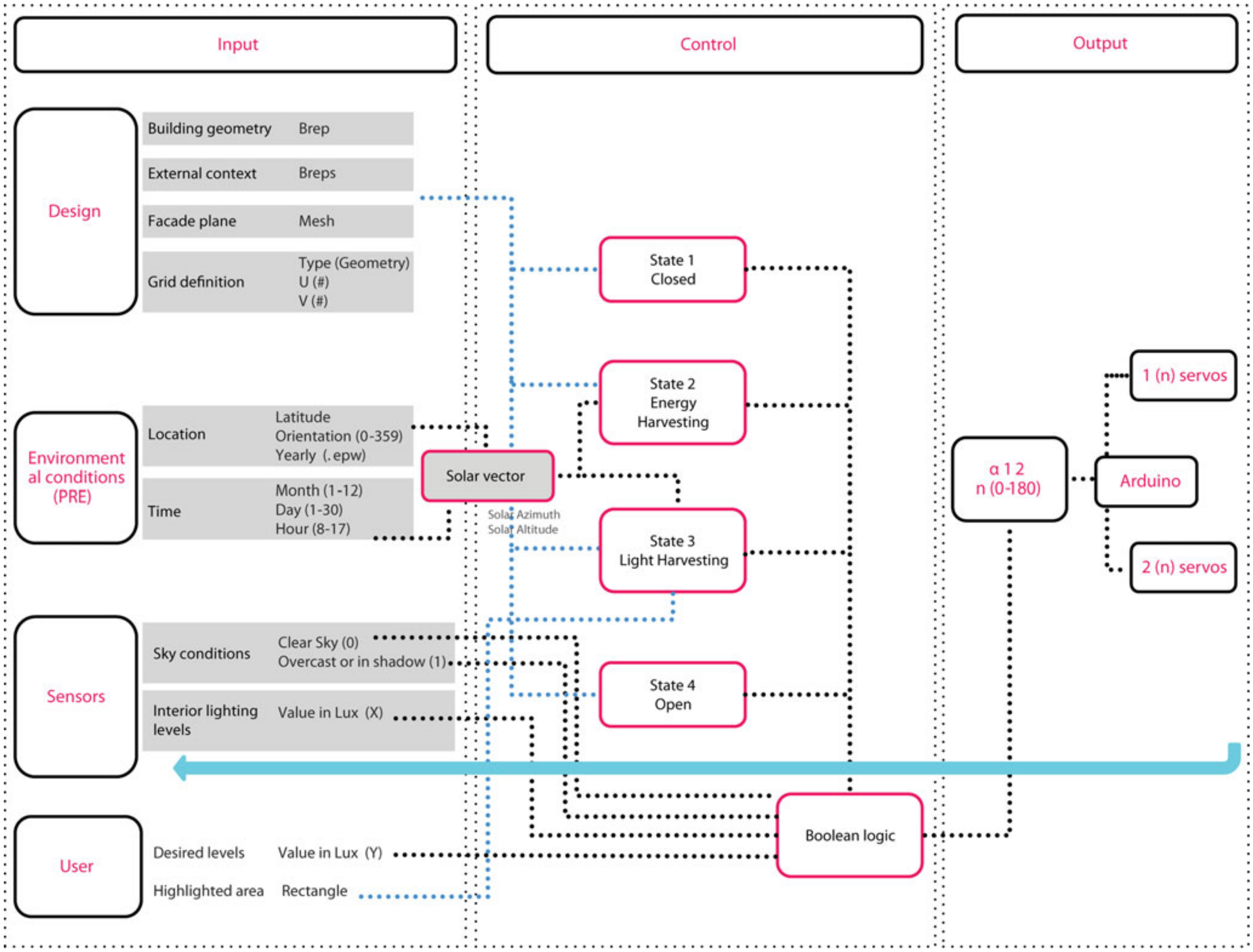
Input
Control
Output
Design
Building geometry Brep
External context Breps
Facade plane Mesh
Grid definition Type (Geometry) U (#) V (#)
Environmental conditions (PRE)
Location Latitude Orientation (0-359) Yearly (.epw)
Time Month (1-12) Day (1-30) Hour (8-17)
Solar vector
Solar Azimuth
Solar Altitude
Sensors
Sky conditions Clear Sky (0) Overcast or in shadow (1)
Interior lighting levels Value in Lux (X)
User
Desired levels Value in Lux (Y)
Highlighted area Rectangle
State 1 Closed
State 2 Energy Harvesting
State 3 Light Harvesting
State 4 Open
Boolean logic
α 1 2
n (0-180)
Arduino
1 (n) servos
2 (n) servos

Fig. 3. Logic diagram (Author).

and cardinal position; the geometry and surface definition (in terms of reflectiveness) of the context objects; the facade plane and its subdivision in components, including a geometrical pattern and its frequency (number of divisions in its local U and V directions); and surface definition.

2.1.2. Predictable environmental conditions data

The resulting input of this category is the sun position in space, but in order to find it, two sets of data are required: site location in terms of latitude and time defined by hour, day, and month.

2.1.3. Sensor gathered data

There are two sets of sensors in the system, the external and the internal ones. The external sensors are situated on each of the faces of every component to define the amount of light that is actually being received, something unpredictable due to cloud conditions. The internal ones are situated on the ceiling of the interior space, to measure the amount of reflected light according to a particular configuration of the dynamic facade components.

2.1.4. User defined data

This implies a further definition of the two main requirements of architectural lighting, namely, the amount and distribution of light in the space. The first one is defined as the preferred intensity in lux, and the second one by selecting a highlighted area for reflections.

2.2. Control definition

The control phase involves the construction of a solar vector from the predictable environmental conditions (location and time), and following that, the design data (facade plane and grid) should be also processed to reach one of four possible states at each panel (Fig. 4).

1. *Closed state:* When a component shares the same plane with the facade, it is at its closed position. Here no light (either direct or indirect) is allowed to the interior, whereas the external surface (PV) would partially absorb it and convert it into energy.
2. *Energy harvesting (solar tracking):* When a component plane (facing its PV side) is perpendicular to the

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

Cell positions

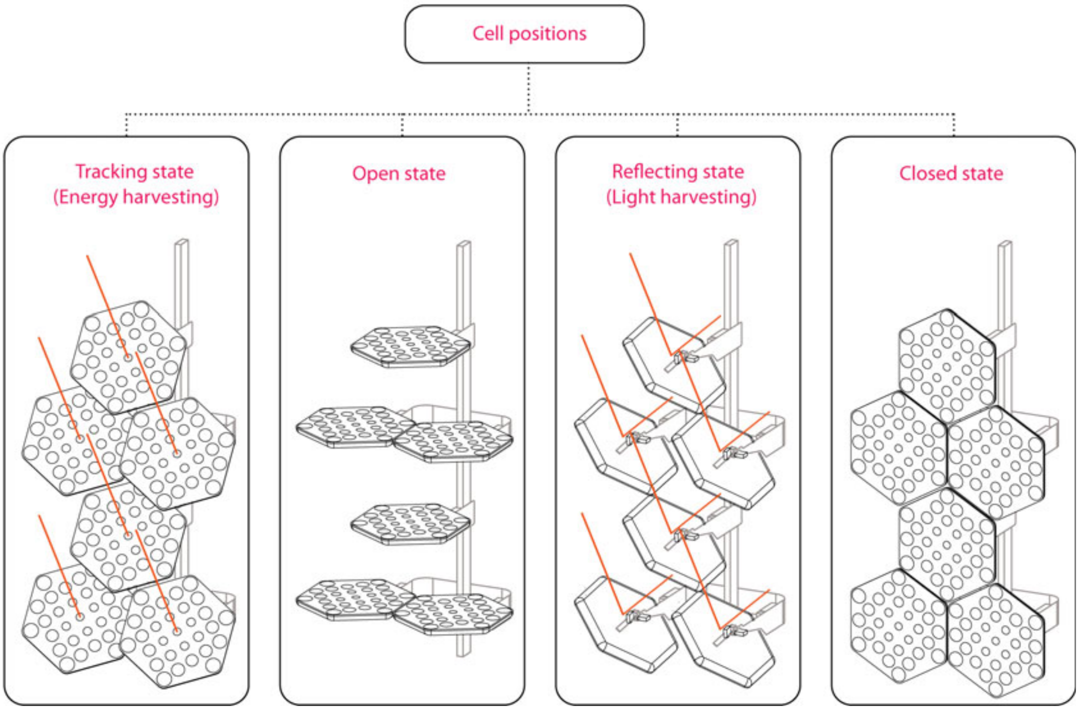

Fig. 4. Panel states, that is, generic positions.

previously defined solar vector, it is at its energy harvesting position. Here the maximum amount of radiation is received by the component, because reflections are minimized, while still allowing some amount of indirect light and views from the inside to the outside.

3. *Light harvesting (reflecting):* when a component plane is aligned (facing its reflective side) at the bisector of the solar vector and the vector linking its centroid to the light target on the ceiling of the interior space, it is at its light harvesting state. Here the maximum amount of light is directed to the interior space, also allowing for views to the outside.
4. *Open:* When a component plane is perpendicular to the facade plane (by default vertical), it is at its open position. Here views from the interior are maximized, because the panels allow for unrestricted and continuous horizontal sights.

2.3. Output data for mechanical and electronic implementation

Finally, the output phase involves the transfer of information regarding the particular states of each component as angle values to run the servomotors embedded in each unit. For that, a signal with information regarding the movements and a source of energy are required.

3. DESIGNER EXPLORATIONS VIA GEOMETRY-BASED GRAPHICAL PROGRAMMING USING BOOLEAN LOGIC (IMPLEMENTATION PHASE 1)

The first implementation of the logic was developed within Grasshopper (http://www.grasshopper3d.com), a graphical algorithm editor integrated to Rhinoceros3D (http://www.rhino3d.com), a three-dimensional modeling software. In this way, the designer had complete control over changes in the design inputs (purely geometrical) and, at the same time, could see the implications of those changes in the performance of the system. This was possible thanks to the use of light tracing simulation tools integrated to the graphical editor: for simulation purposes Radiance (http://www.radiance-online.org/), via Honeybee (an add-on to the editor; http://www.grasshopper3d.com/group/ladybug), was mainly used. The implementation details will be discussed below.

3.1. Input definition

3.1.1. Design

Building geometry and context. The first step in the definition involves the creation of base geometries simulating the actual conditions of the location and the particular space or building that the system is going to cover (Fig. 5). For this example, we built a room with the common dimensions of an individual office space (5 × 5 × 3 m width × depth × height) to work as a

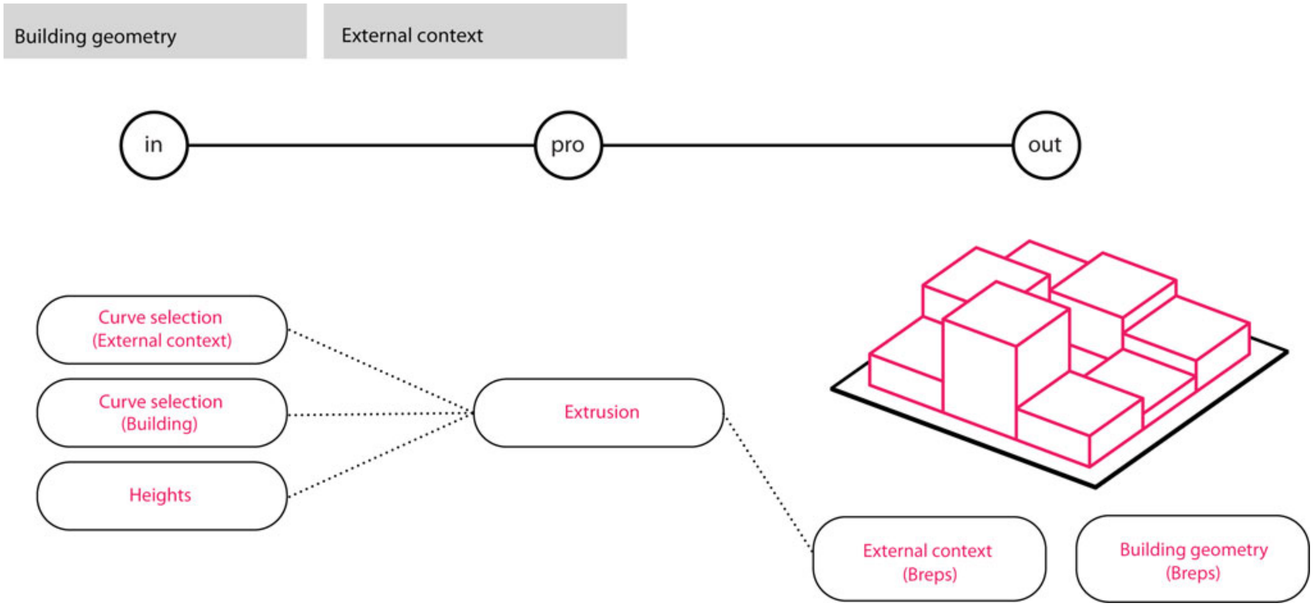
Building geometry
External context
in
pro
out
Curve selection (External context)
Curve selection (Building)
Heights
Extrusion
External context (Breps)
Building geometry (Breps)

Fig. 5. Building and context parameters.

testing rig in future daylight simulations of the system. The construction of these geometries could be either parametric definitions in GH or fixed entities brought from Rhino.

Facade plane and grid definition. Having defined the general geometry, the next stage required is to choose the plane or planes where the system would be implemented, and subsequently define the base grid for the system, that is, the division to create the planar components (perimeters) that would rotate with two degrees of freedom (Fig. 6). Following from that, it was also necessary to define the area centroids of all elements, because these would be used as the centers of

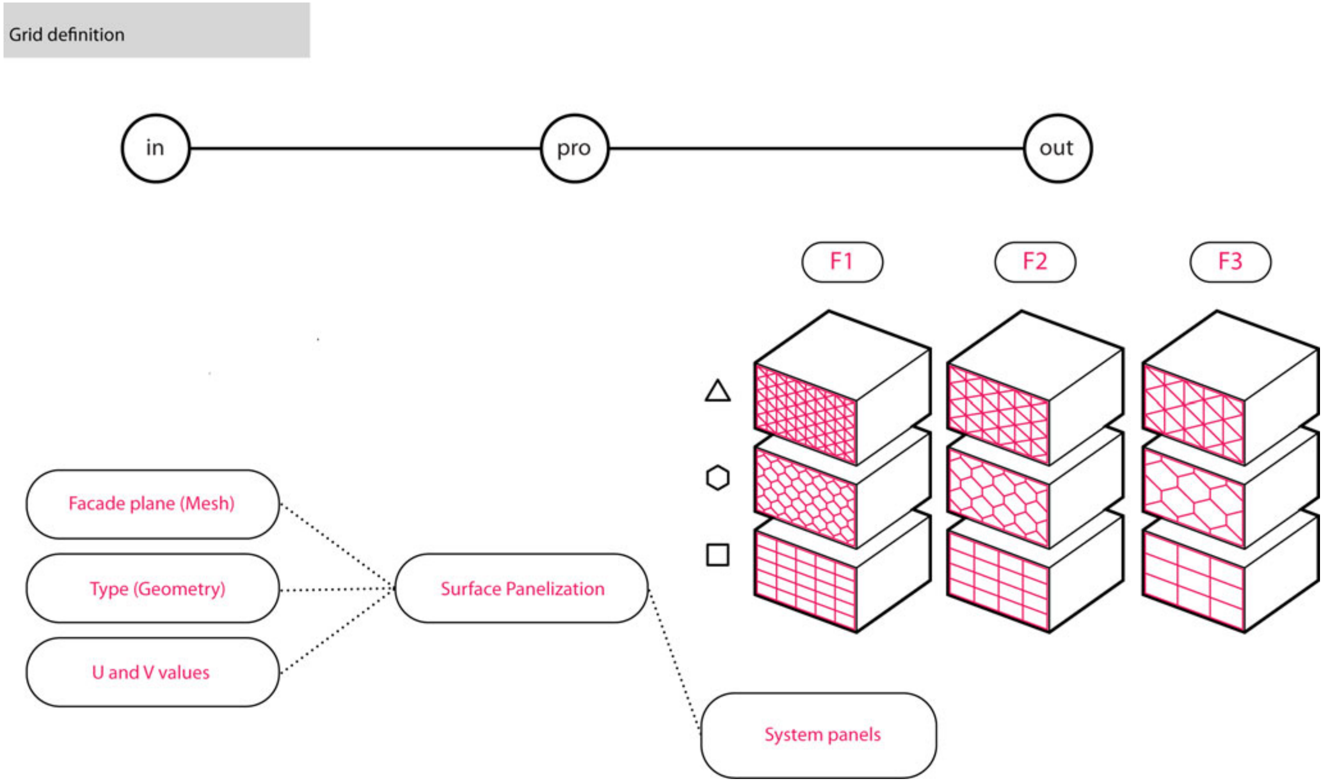

Fig. 6. Grid definition.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

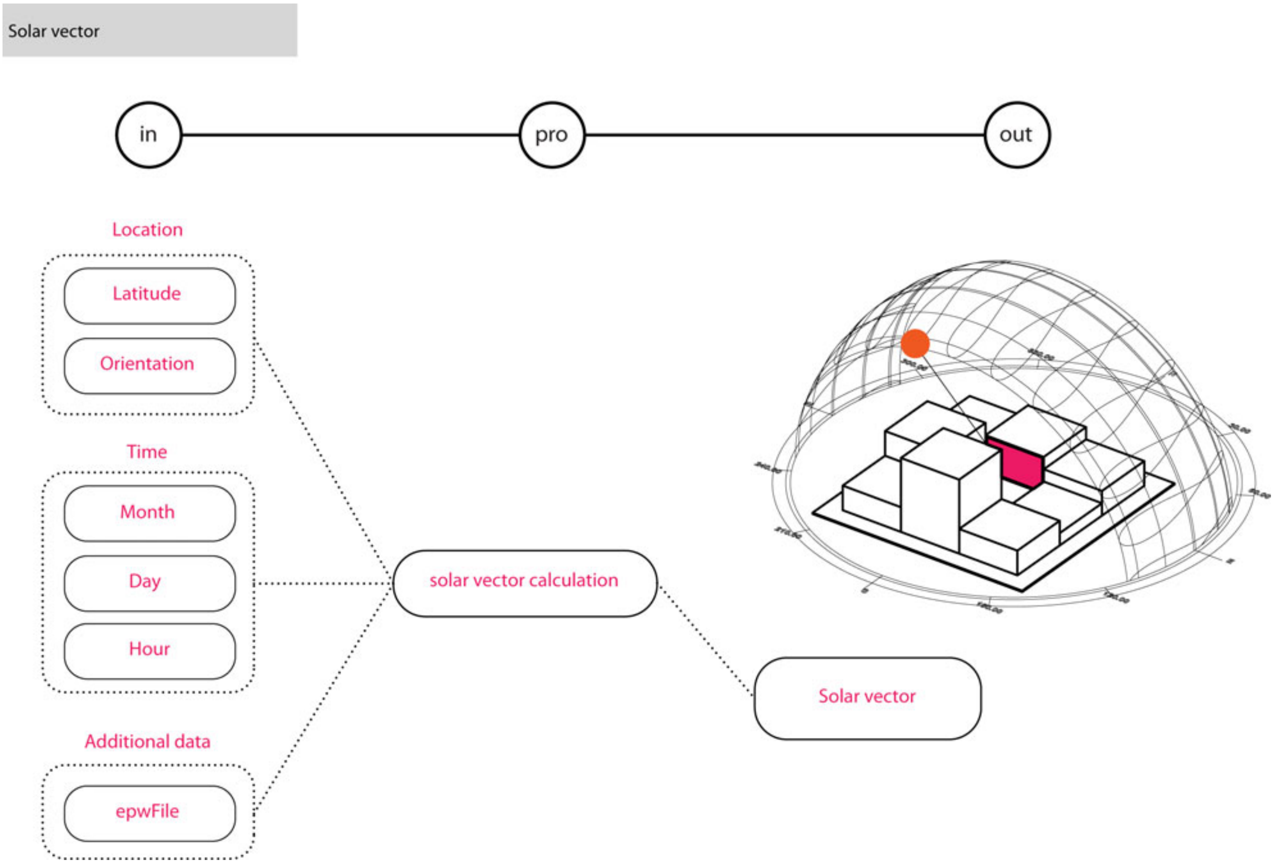

Fig. 7. Solar vector.

rotation. In this case, predefined triangular, square, hexagonal, and rhombic division mechanisms provided by the plugin Lunch Box (http://www.grasshopper3d.com/group/lunchbox) were employed, but any other coded or handmade subdivision may be implemented, including irregular ones.

3.1.2. Predictable environmental conditions

Location and time. In this next group of inputs, the created geometries were to be placed in a solar context, and for that, three sets of information were needed: data regarding the geographical location, relative orientation of the whole geometry (context constructions and analysed building) in that location, and the specific time at which the calculations should take place (Fig. 7). Geographical location data and time are used to calculate the solar vector, in this case using the capabilities provided by the components from the Grasshopper plugin LadyBug (http://www.grasshopper3d.com/group/ladybug). An .epw file for the particular location is

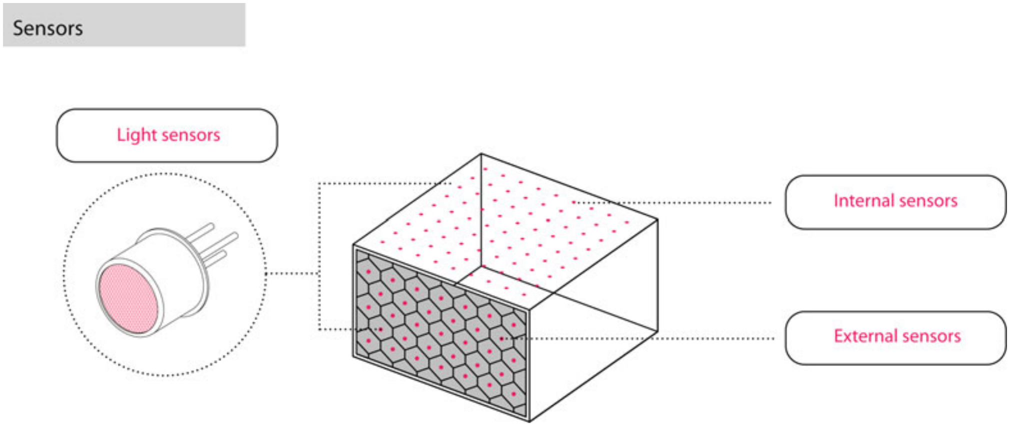

Fig. 8. Sensor points for evaluation.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

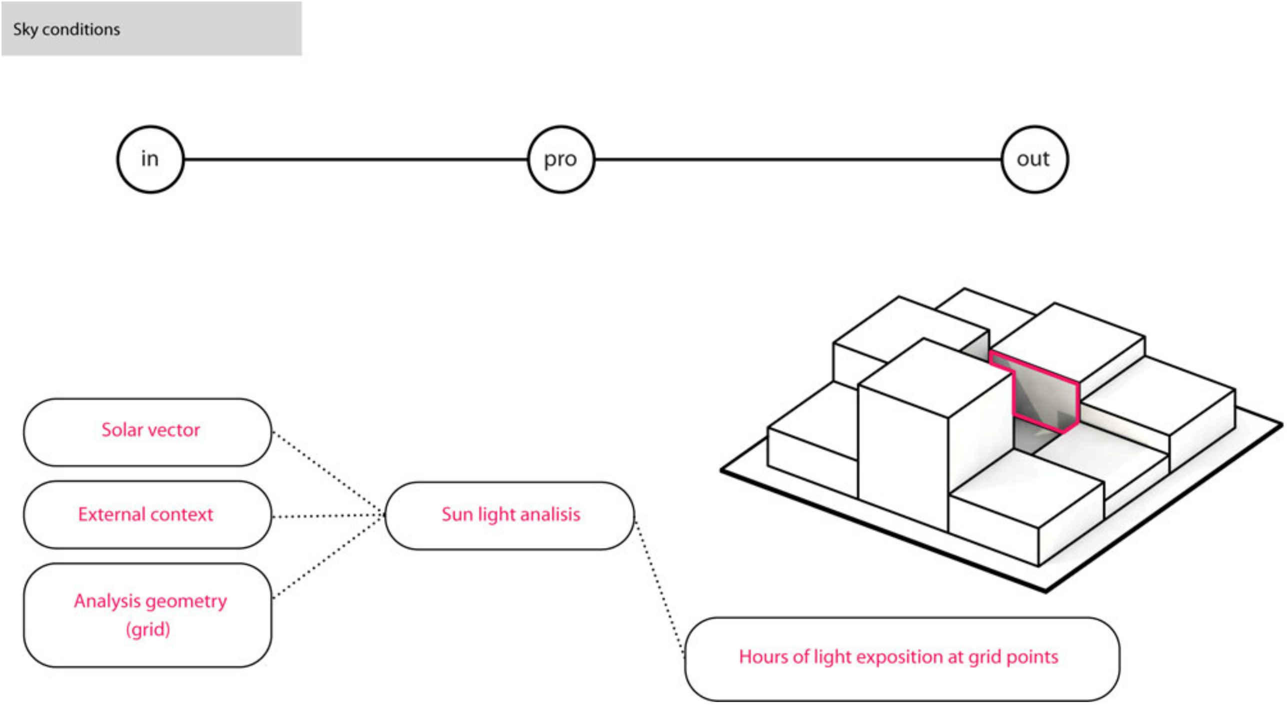

Fig. 9. Sky conditions.

also provided, which contains average annual environmental information on an hourly basis. This would be required for simulation processes implemented later on in this same definition (Fig. 8).

3.1.3. Sensors

Sky conditions. Sky conditions define whether there is direct light incidence or not for each panel of the system (Fig. 9). The cases where no incidence would happen can be twofold, due to either cloud cover (unpredictable) or obstruction by other existing masses (context). In a real implementation, this information would be provided by light sensors at each panel, but in order to make this computational definition workable without the need of a physical implementation, virtual points for simulation were used, meaning that this simulation was to be limited to the evaluation of shadows projected by context. The component to evaluate if there is visible sun from each point at a particular time was also provided by the Ladybug plugin.

Lighting levels. Internal lighting levels are simulated using Radiance via Honeybee (Fig. 10). The simulation requires information regarding the objects in the context, its surface qualities (material), and an .epw file with the average hourly weather data of a particular location. The simulation is run for a specific time, and it gives back lux levels for the points evaluated. Here, 27 points on a grid at a 0.7-m height equivalent to desk level were evaluated. The results are visually represented by a color gradient according to the relative value of the points evaluated.

3.2. Control

Having defined the four states at which each component (rotating plate) can be positioned, a Boolean logic based on information given by sensors and the user will be evaluated in the following way:

> Given that FOR Sky conditions and shadow: 0 = no direct light, 1 = direct light, AND Lighting levels inside = X, User lighting levels = Y:
>
> If X > Y: 0 = State 1 and 1 = State 2 Else: 0 = State 4 and 1 = State 3.

In other words, if the existing lighting levels (as measured by interior light sensors) are higher than those selected by the user, then the panels that have no direct solar light incidence will be closed and those receiving insolation will be positioned perpendicular to the solar rays, so PV cells will absorb energy, blocking the sun inside but allowing views. If the opposite is true, that is, the values for the actual lighting levels inside are lower than those required by the user, then, if the panels are in shadow, they will be positioned as open, whereas if they receive direct light, they will be positioned in a reflecting way as to increase the interior lighting levels (Fig. 11).

This logic will be repeated *n* times, *n* being the number of individual components (panels). In the process, changes in the positioning of each panel will be affecting the lighting levels measured by the interior sensors, making it work in an iterative loop. This, however, is not implemented in the Grasshopper definition for two reasons, first, the processing time of each simulation is

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

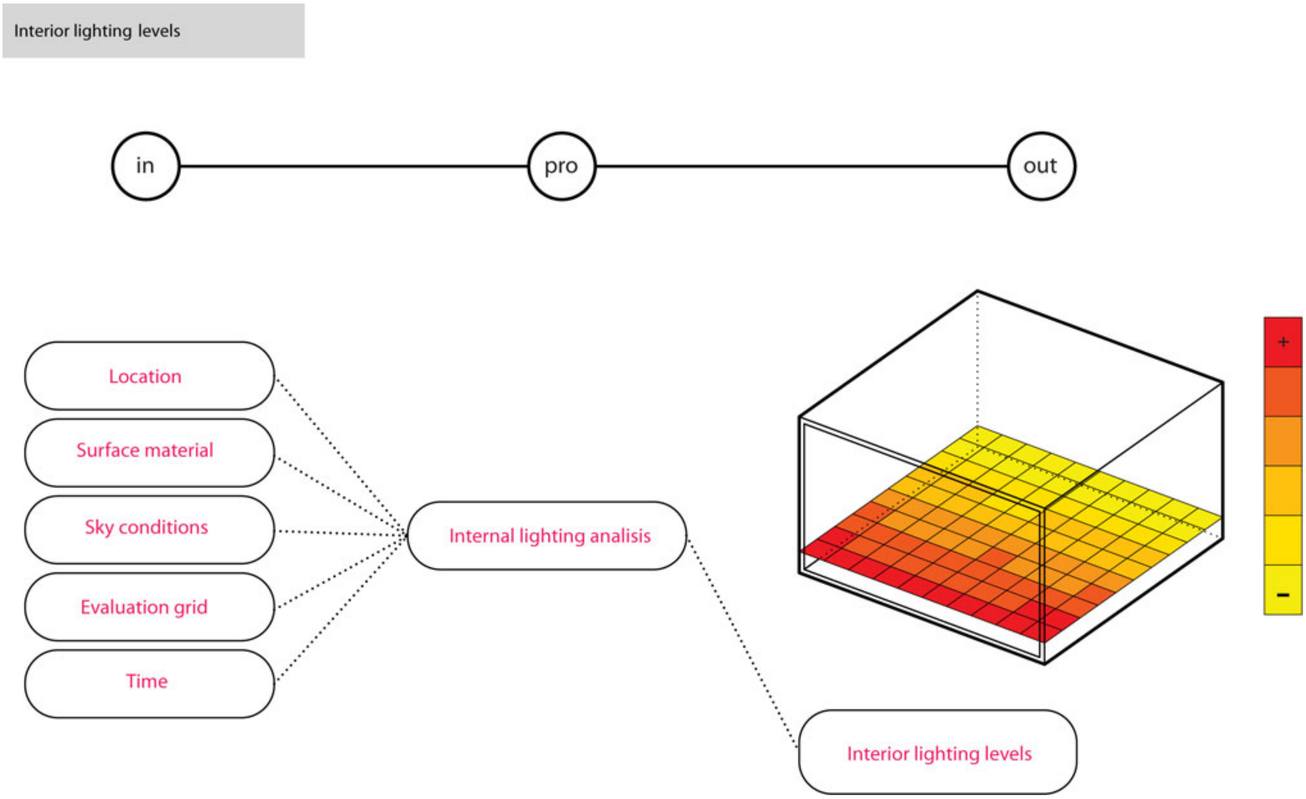

Fig. 10. Interior lighting levels.

about a minute, implying an hour per iteration with the actual 53 panels model, and second, the selected platform for simulation does not allow sequential time conditions change in a single run.

3.3. Outputs

The final phase of the process implies the transmission of the information regarding the particular positioning of each panel to the actuators (servomotors) that will be making the movement effective. For that, the information will have to go out from Grasshopper and be received by an external microprocessor, in this case an Arduino (http://www.arduino.cc/), that transforms such numeric information into a pulse-width modulaton signal understandable by the Servomotor. The process is carried out by Firefly (http://www.fireflyexperiments.com/#home), an already developed plugin for Grasshopper. In this definition, the lists of values ($n = 53$ here) are being filtered on a panel basis, using two servomotors per unit (Fig. 12).

In terms of information transference, the controlling unit (PC) sends the angles to each servomotor, but in order for the motors to understand the information, it has to be converted into a pulse-width modulation signal by the Arduino microprocessor. In addition, the information captured by the light-dependent resistor photocell has to go back to the processing unit via the same microprocessor. In terms of the power supply for the movement, each component has two PV cells, each one with the capacity to provide for the functioning a single servomotor unit; however, there will not be constant energy production by the PV cells, either because they might be under shadow or due to the components being in State 3, that is, light harvesting mode, where the PVs are facing downward. Because of that, a battery is provided, harvesting energy for later use. In this first implementation, the system requires the inclusion of a comparator that evaluates whether the PV cells are providing the required power or not (6 V), and accordingly would send a 0 or 1 logic signal to the Arduino. Depending on the case, the Arduino would then activate one of the two relays so the servomotors are powered directly by the PV cells or by the battery.

4. USER EXPLORATIONS VIA STANDARD PROGRAMMING SUPPORTED BY THE INCLUSION OF FUZZY LOGIC (IMPLEMENTATION PHASE 2)

Having defined the design inputs from explorations carried out using the previously described process, a following step comprised the development of standard programming to control the system at an operational (real-life) stage. In developing the system, however, it was noticed that Boolean rules would not always be appropriate due to the highly subjective requirements of users in terms of quantity and quality of lighting. Following from that, a further approach based on fuzzy logic was implemented to support the operation of the system, starting with the expansion in the amount of data captured by the external and internal sensors, and

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

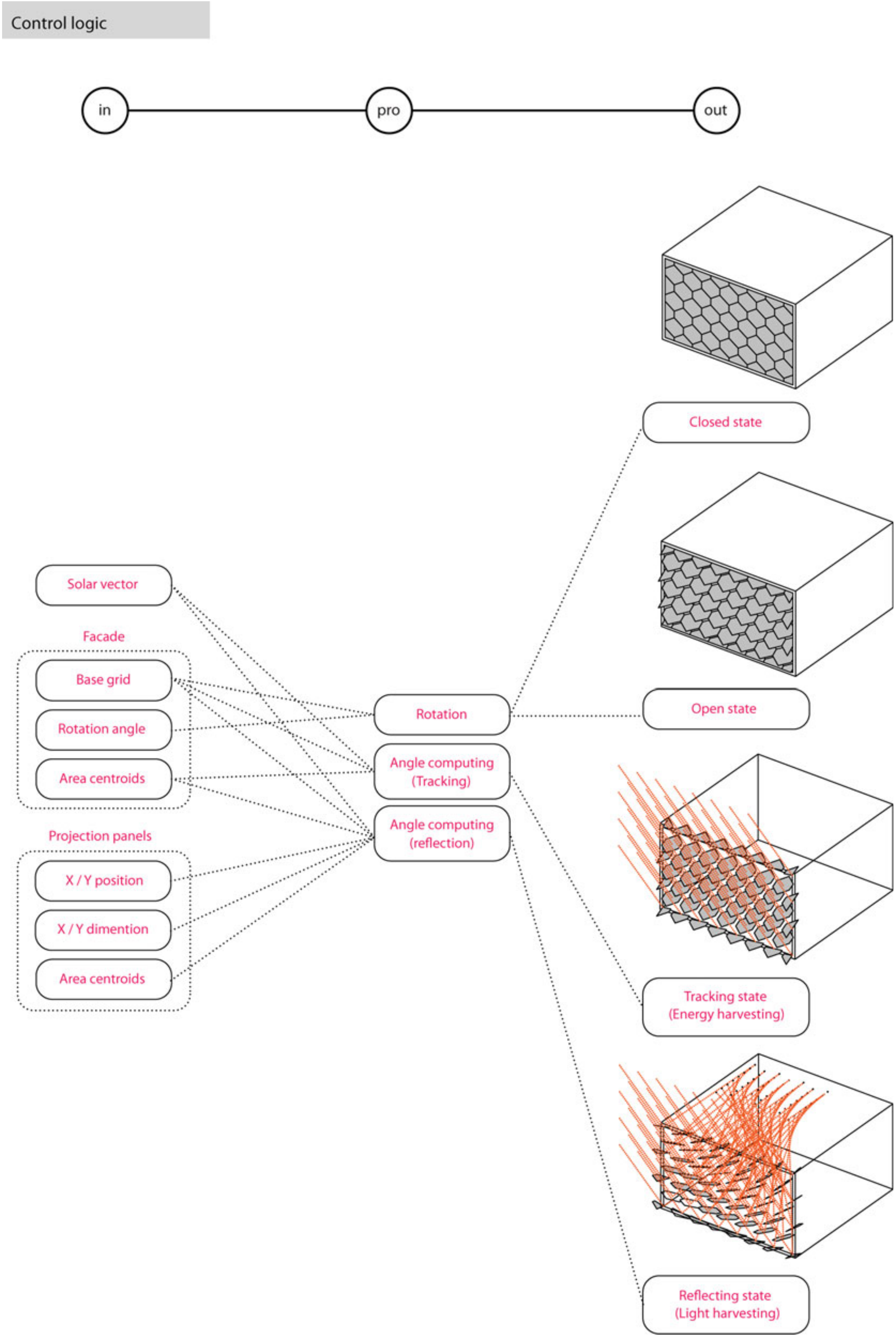

Fig. 11. General view of states.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

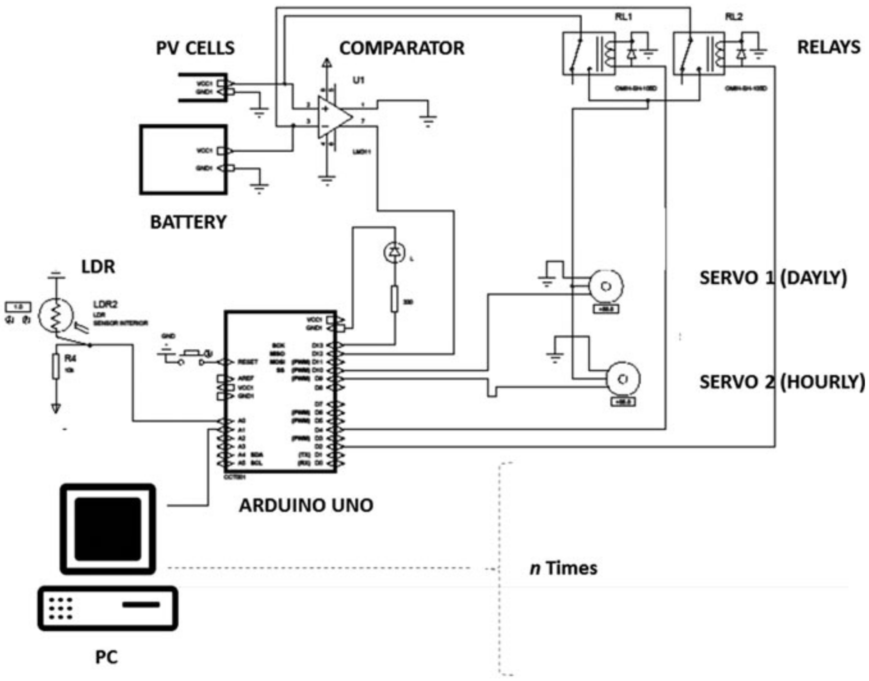

Fig. 12. Electronic diagram (Author).

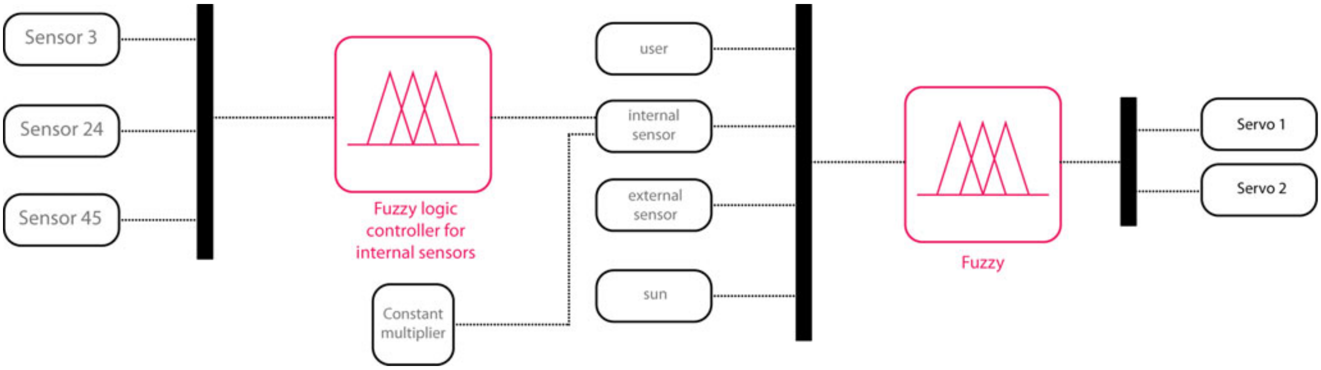

Fig. 13. Fuzzy diagram.

adding the possibility for constant variability in the amount of light set by the user.

Fuzzy logic concepts can be used to translate into mathematical terms some imprecisions of linguistic rules sets (Fig. 13). Using expert knowledge this can allow for the creation of automatic control strategies (Zadeh, 1965; Gomide et al., 1995). This general technique presents a design methodology, but for the development of this system, the method proposed by Lee (1990) was used. The result is a rule-based inference system in which a fuzzy theory set and fuzzy logic create the tool to solve this type of problems (Tanscheit, 1999).

Fuzzy logic was implemented at two stages in phase. First, it was applied to define the internal daylight levels from values given by three sensors (or evaluation points for digital simulations) at different depths in the room, because averaging such values could be a misleading technique to obtain results. As an example, having extremely high values near the facade and rather poor at the bottom, even if resulting in high values in average, may imply the worst case of glare in the space. Second, in this final stage, fuzzy logic was applied to find the positioning of the panels based on four types of inputs: external lighting levels, internal lighting levels (fuzzy results), solar position, and user requirements. The use of fuzzy logic allowed including the input *user preferences*, which was not considered in the first phase previously described.

To implement the logic, membership functions had to be established; these functions have numeric values between 0 and 1. In this application, triangular and trapezoidal membership functions sets were employed for the fuzzification, to avoid abrupt changes of the inputs (from low to high luminosity) with values defined for each kind of input as described below.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

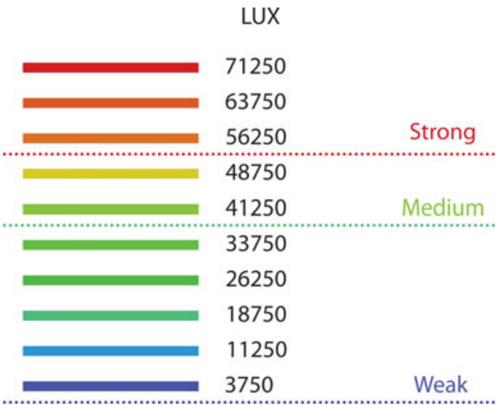

Fig. 14. External sensor luminosity scale.

4.1. Inputs

4.1.1. *External sensors*

The external sensors are located in each panel and operate with values between 3750 and 71,250 lux. Within these values, three ranks were created to translate the sensor value in user requirement language as follows: first rank (weak light) with values between 3750 and 37,500 lux, second rank (middle light) with values between 37,500 and 45,000 lux, and third rank (strong light) with values between 45,000 and 71,250 lux or more (Fig. 14). In the fuzzy logic, the external sensor input has the implementation shown in Figure 15.

4.1.2. *Internal sensors*

Internal sensors were adjusted to operate using values between 100 and 2850 lux, for which three ranks were also defined: weak light, middle light, and strong light in accord with the luminosity shown in Figure 16.

As mentioned previously, however, this input includes its own fuzzy control (Fig. 17), implying that each sensor has four states: very dark (between 0 and 280 lux), dark (between 280 and 1260 lux), bright (between 1260 and 2000 lux), and very bright (between 2000 and 2850 lux). Using the rules shown in the Table 1, it was possible to compare the three sensors and know the luminosity in all rooms (Fig. 18).

Using the rules, the output shown in Figure 19 was obtained. These results are to be used as input to the second application of fuzzy control.

4.1.3. *User preferences*

User light preferences are set by pushing the plus or minus buttons at the user interface. They have the implementation in fuzzy logic as demonstrated in Figure 20.

4.1.4. *Solar position*

Solar position is given by azimuth and elevation angles calculated for a specific location and time.

4.2. Control

In this implementation, the method of center of areas was employed to determine the servomotors positions (tracking, reflective, open, or closed), taking into account the input parameters and their evaluation. The fuzzy outputs are calculated in the fuzzy inference according to the values shown in Table 2 (Fig. 21).

The fuzzy logic control takes into account the four input ranges that are given, namely, the user preferences, internal sensor, external sensor, and position of the sun. An example of a rule to generate the output angles for the two servomotors would be the following: if the user wants more light, the internal sensor is in the low range and the external sensor is in the high range, then the servomotors should be in the reflective position, and that would be solved according to the actual solar position and target point inside (here by default corresponding to a projected point on the ceiling). The implementation of fuzzy logic was designed using the fuzzy toolbox of MATLAB, where relationships were defined according to Table 3.

Following the rules for the fuzzy inference system, the graphics that represents the functions relevant to the desired outputs are shown in Figure 22, corresponding to the tracking and reflective states according to the tests.

This logic is able to create a dynamic system responding to the changing position of the sun, the changing user require-

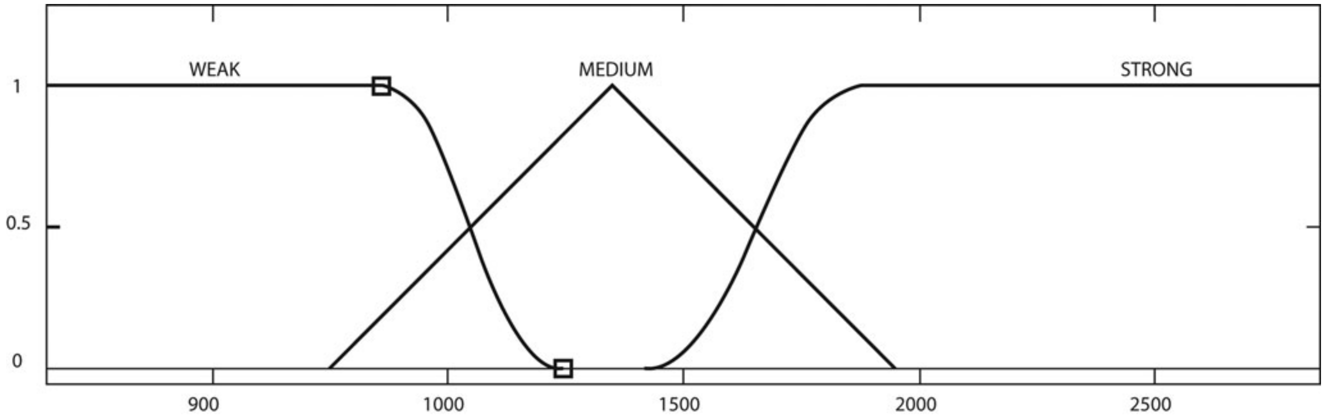

Fig. 15. External sensor fuzzy logic input.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

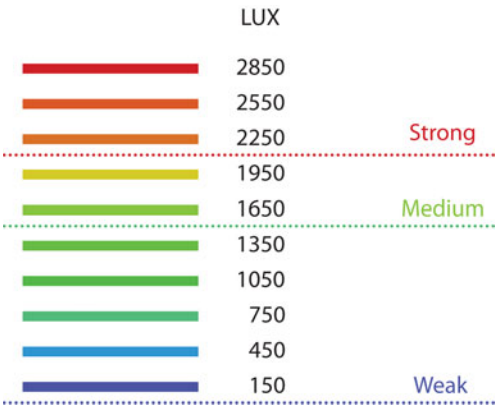

Fig. 16. Internal sensor luminosity scale.

ments, and the changing amounts of exterior and interior light, creating a programmable logic controller system. This logic can be replicated for *n* panels of the prototype, creating an independent system where the user can become a codesigner in real time.

4.3. Outputs

4.3.1. Daily servomotor

The day-based servomotor can have values between 0° and 180°, allowing for three different positions: between 0° and <89° where PV surface is being active, at 90° where it is closed, or between >91° and 180° where it acts as reflective.

Table 1. *Fuzzy logic rules to the internal sensor*

Sensor 3	Sensor 24	Sensor 45	Luminosity
VB	VB	VB	Bright
VB	VB	B	Bright
VB	VB	D	Normal
VB	VB	VD	Normal
VB	B	VB	Bright
VB	B	B	Bright
VB	B	D	Normal
VB	B	VD	Normal
VB	D	D	Dark
VB	D	VD	Dark
VB	VD	D	Dark
VB	VD	VD	Dark
B	VB	VB	Bright
B	VB	B	Bright
B	VB	D	Normal
B	VB	VD	Normal
B	B	VB	Bright
B	B	B	Bright
B	B	D	Normal
B	B	VD	Normal
B	D	D	Dark
B	D	VD	Dark
B	VD	D	Dark
B	VD	VD	Dark
D	D	D	Dark
D	D	VD	Dark
D	VD	D	Dark
D	VD	VD	Dark
VD	D	D	Dark
VD	D	VD	Dark
VD	VD	D	Dark
VD	VD	VD	Dark

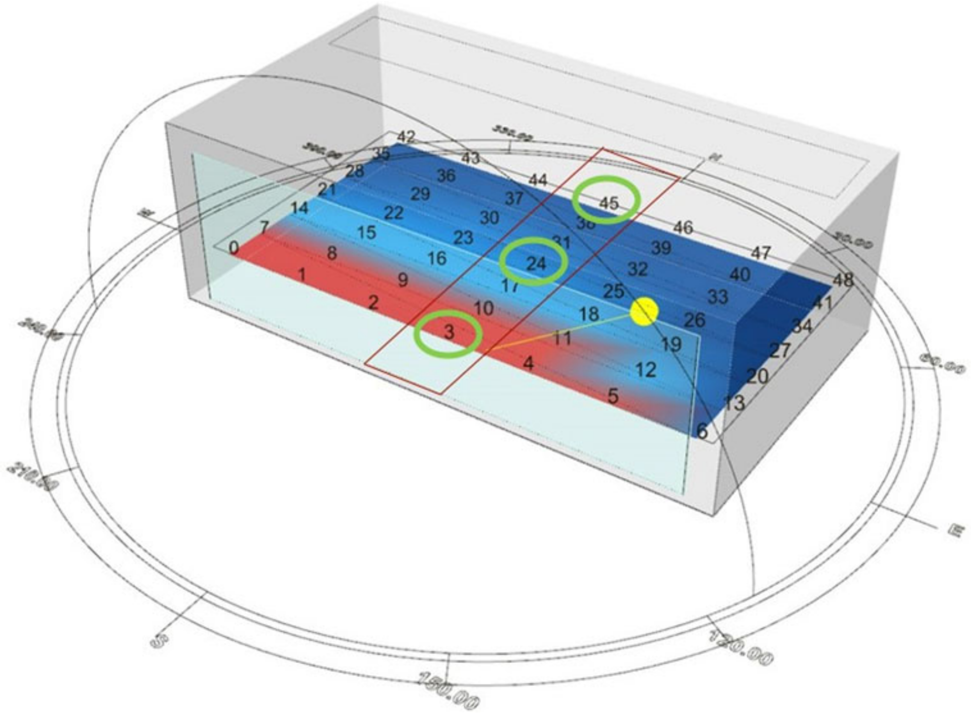

Fig. 17. Simulation points.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

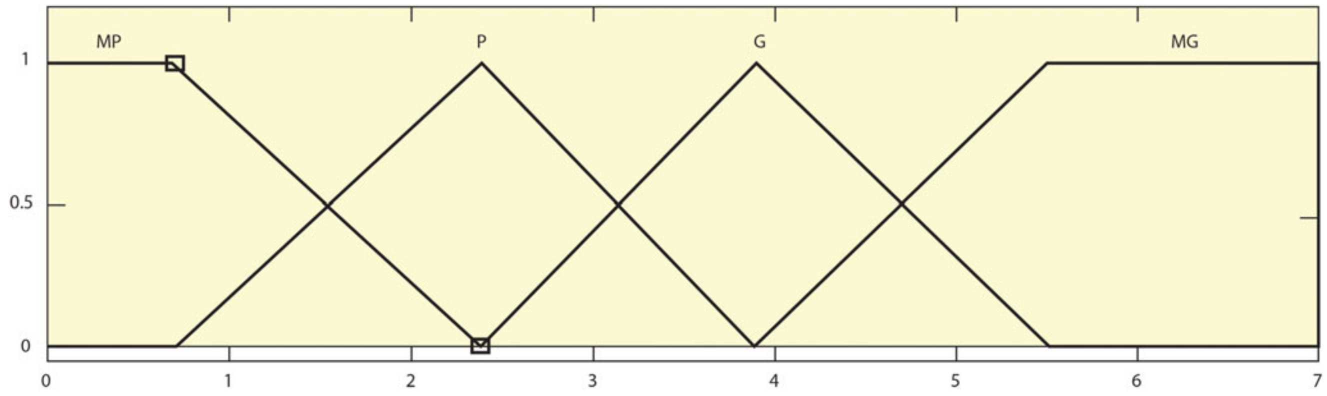

Fig. 18. Fuzzy logic inputs.

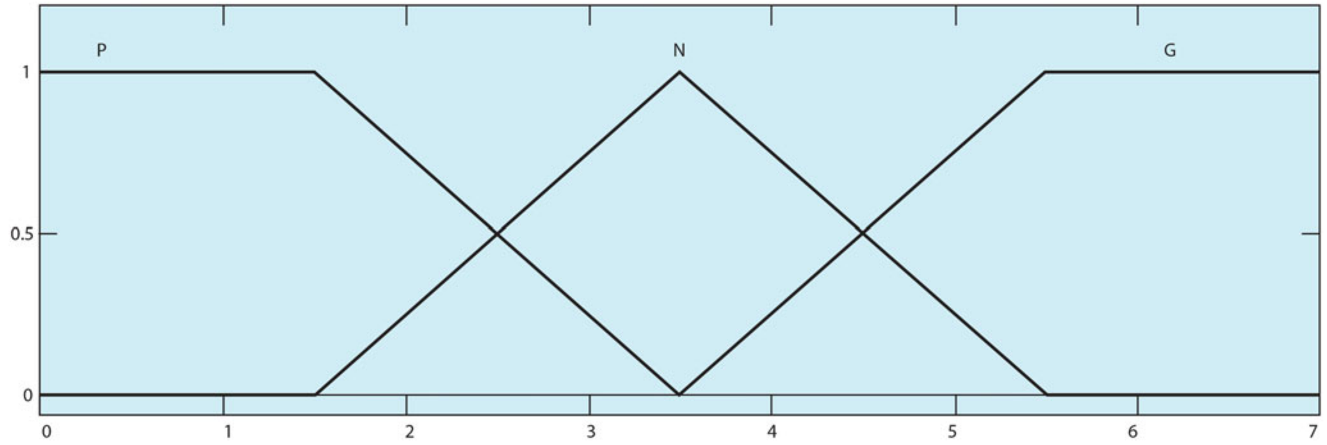

Fig. 19. Fuzzy logic output of internal luminosity.

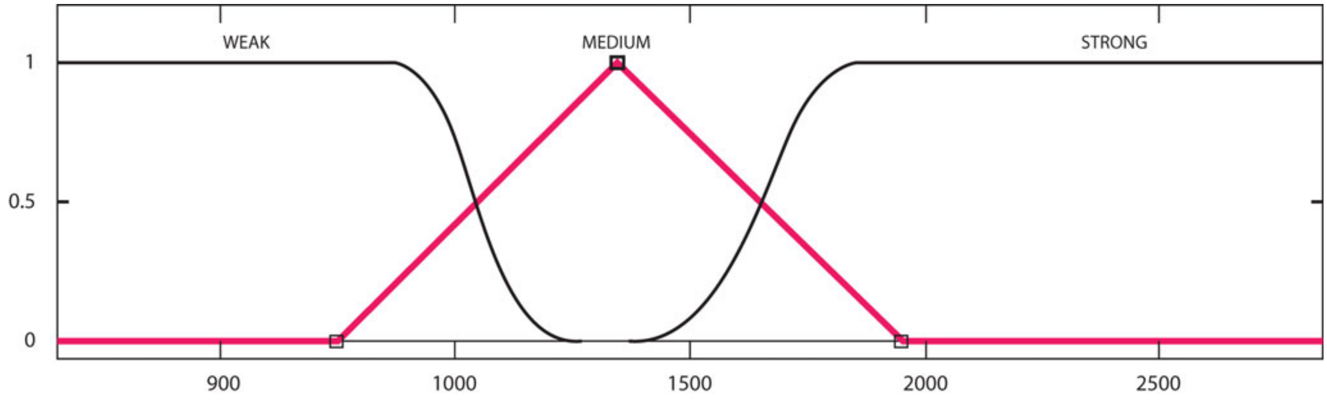

Fig. 20. User requirement fuzzy logic input.

Table 2. *Fuzzy outputs*

Open	Reflective	Closed	Tracking
0	0.75	0.5	0.25

The daily servomotor has the implementation in fuzzy logic as shown in Figure 23.

4.3.2. *Hourly servomotor*

The hour-based servomotor uses the complementary value to the azimuth angle to be able to follow the sunrays at any

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

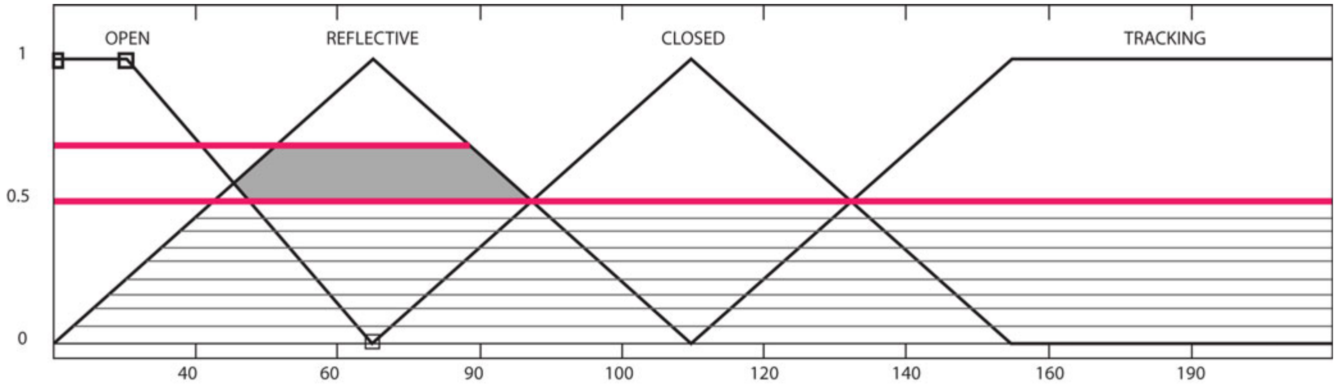

Fig. 21. Membership function of outputs cut.

hour of the day. The hour servomotor has the implementation in fuzzy logic that is demonstrated in Figure 24.

As implemented in Phase 1, the resulting positioning angles are transmitted to the servomotors via an Arduino microcontroller. Here the MATLAB communication toolbox is used for reception, processing, and transmission between the sensors and the servomotors (Fig. 25). The system has the ability to be implemented across a dynamic facade, because the computational consumption for the process is very low, allowing for quick response times, that is, minimum delays not noticeable to the human eye.

However, if the system were to have a commercial implementation, it would have to use a simpler arrangement with a more sophisticated technology, probably a custom programmable logic controller unit controlling all devices within a PROFINET network including wireless communication as shown in Figure 26.

5. EVALUATION OF THE SYSTEM

As initially proposed, the two phases of design were intended to allow for designer and user explorations, respectively. In order to evaluate the performance of the system in both respects, a virtual testing rig of 7-m width, 5-m depth, and 2.5-height was defined. Lighting levels simulations were carried out for each of the 11 h of direct lighting on a south facade for December 21, the date when the sun is at its lowest position in the area where the project was located (Girardot, CO). Applying the same configuration of the proposed light sensors, simulations involved a 6 × 7 node grid, elevated at 0.7 m. The simulations were also carried out using Radiance via Ladybug on the Grasshopper–Rhino platform.

In the first case, regarding designer explorations, genetic algorithms were implemented inside the GH definition using its embedded tool, Galapagos (http://www.grasshopper3d.

Table 3. *Fuzzy logic rules*

	Sun Position			Servomotor		
Inner Sensor	Azimuth	Elevation	User Requirement	Daily	Hour	Dynamic Facade Position
Weak light	θ	β	More light	θ+90°	180°-β	Reflective
Weak light	θ	β	More light	θ+90°	180°-β	Reflective
Weak light	θ	β	More light	0°	90°	Open
Middle light	θ	β	More light	θ+90°	180°-β	Reflective
Middle light	θ	β	More light	θ+90°	180°-β	Reflective
Middle light	θ	β	More light	0°	90°	Open
Hard light	θ	β	More light	θ+90°	180°-β	Reflective
Hard light	θ	β	More light	θ+90°	180°-β	Reflective
Hard light	θ	β	More light	0°	90°	Open
Weak light	θ	β	Less light	θ−90°	180°-β	Tracking
Weak light	θ	β	Less light	θ−90°	180°-β	Tracking
Weak light	θ	β	Less light	90°	90°	Close
Middle light	θ	β	Less light	θ−90°	180°-β	Tracking
Middle light	θ	β	Less light	θ−90°	180°-β	Tracking
Middle light	θ	β	Less light	90°	90°	Close
Hard light	θ	β	Less light	θ−90°	180°-β	Tracking
Hard light	θ	β	Less light	θ−90°	180°-β	Tracking
Hard light	θ	β	Less light	90°	90°	Close

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

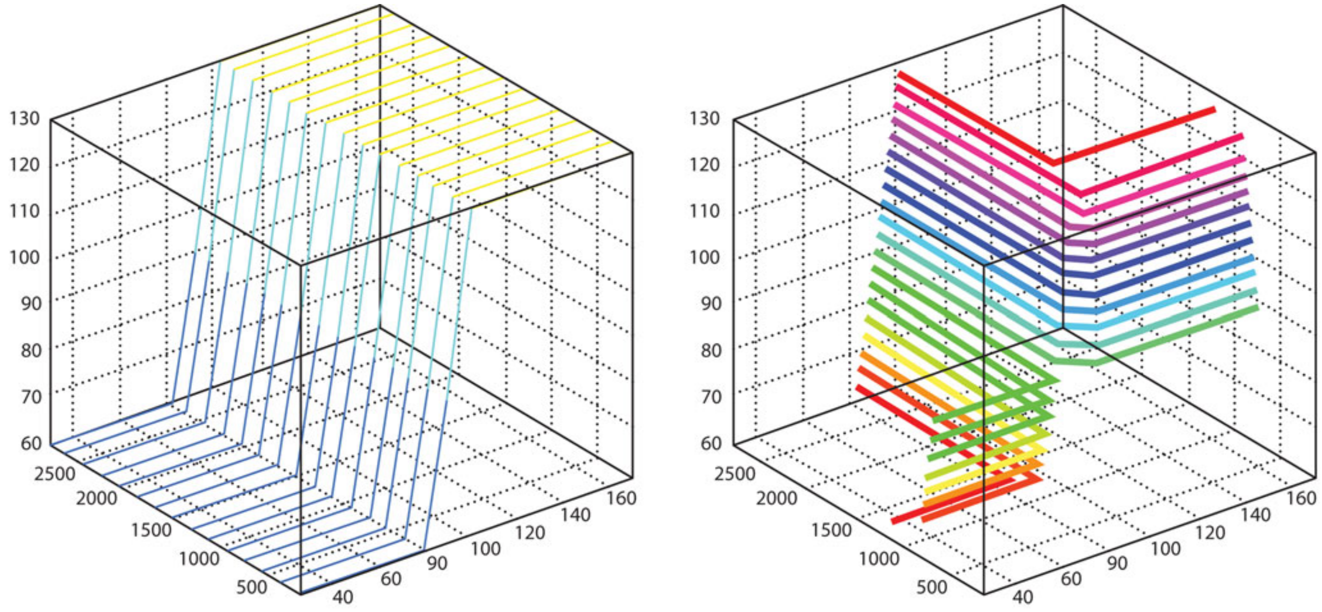

Fig. 22. Representative surface generated by the rules.

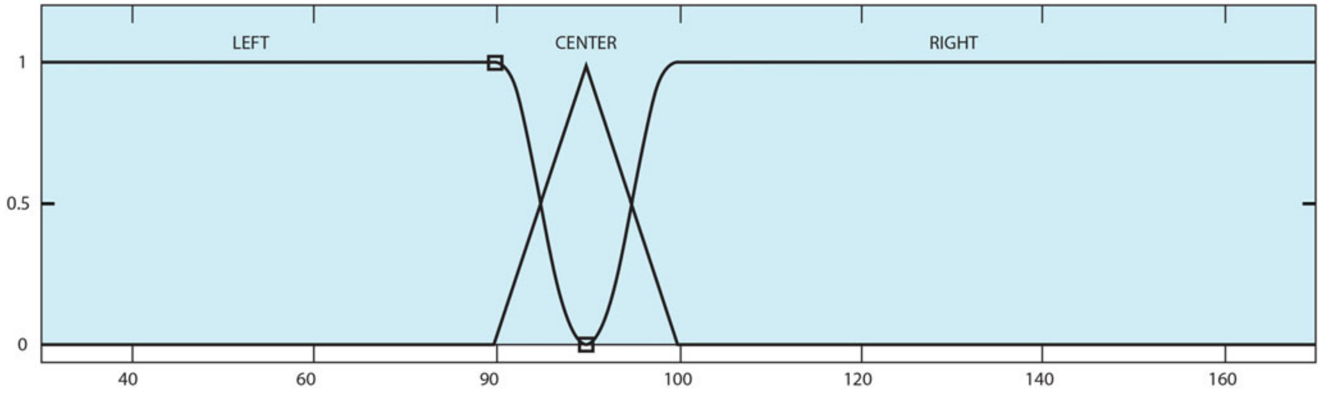

Fig. 23. Daily servomotor fuzzy logic output.

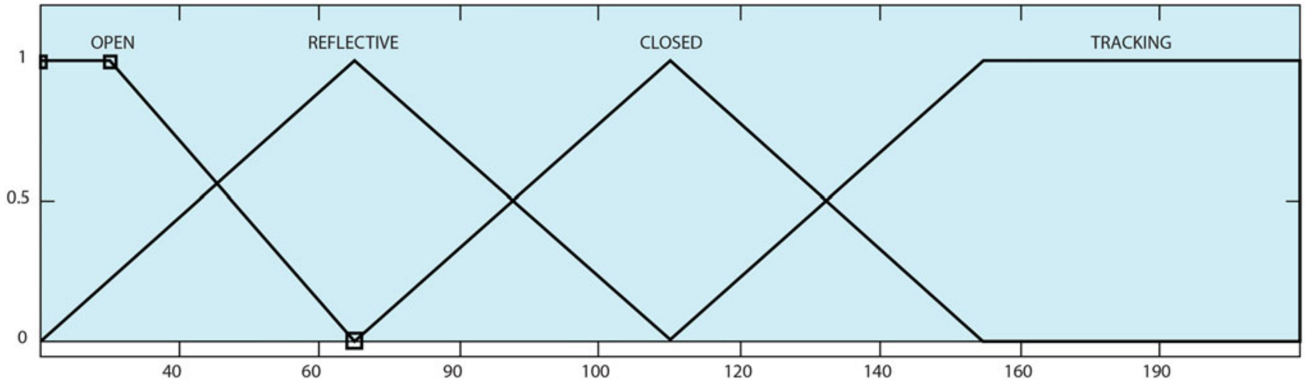

Fig. 24. Hour servomotor fuzzy logic output.

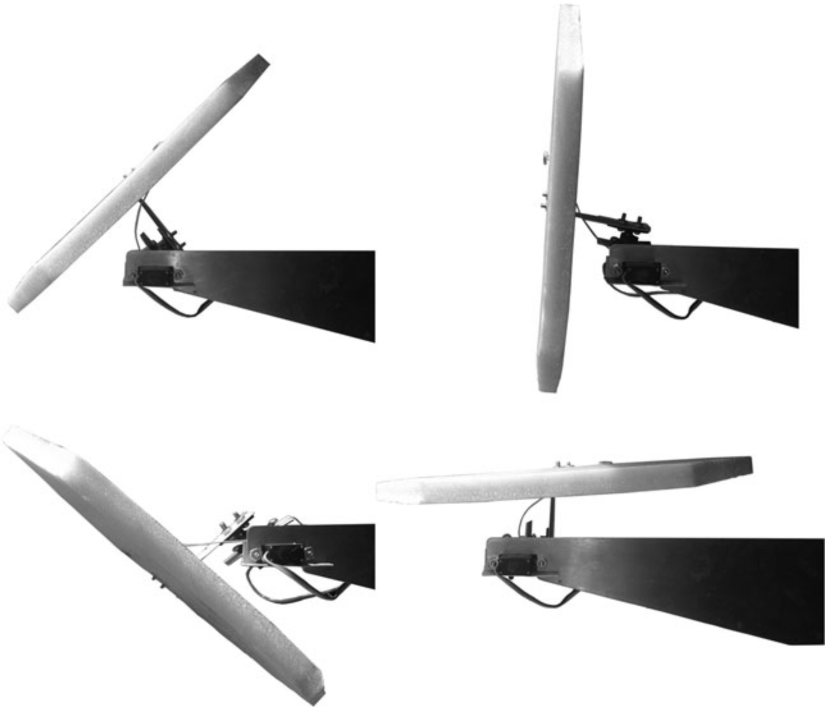

Fig. 25. Prototype using MATLAB control.

com/group/galapagos), to explore the performance of variable options of the grid definition (geometry and ultraviolet frequencies) and to choose optimal configurations. The options explored included three types of regular grid geometries (triangular, square, and hexagonal) and frequency ranges of 5 to 10, in U (equivalent to element lengths of 500–250 mm) and 10 to 20 in V (equivalent to lengths of 700–350 mm). These sizes were considered viable for implementation in real life, because smaller panels would imply too many servomotors (and thus high possibilities for failure), while bigger ones would imply important loads of the panel (and thus the requirement for high-demand servomotors), as well as important space requirements to move inside the covered areas. Though the results favored hexagonal grids with high length ranges, the actual performance in terms of lux values and the curves of penetration did not vary significantly among the evaluated possibilities, with figures below 50 lux as maximum average difference in both cases.

More significant were the results of simulations to test user requirements, where we tested subjective demands for less or

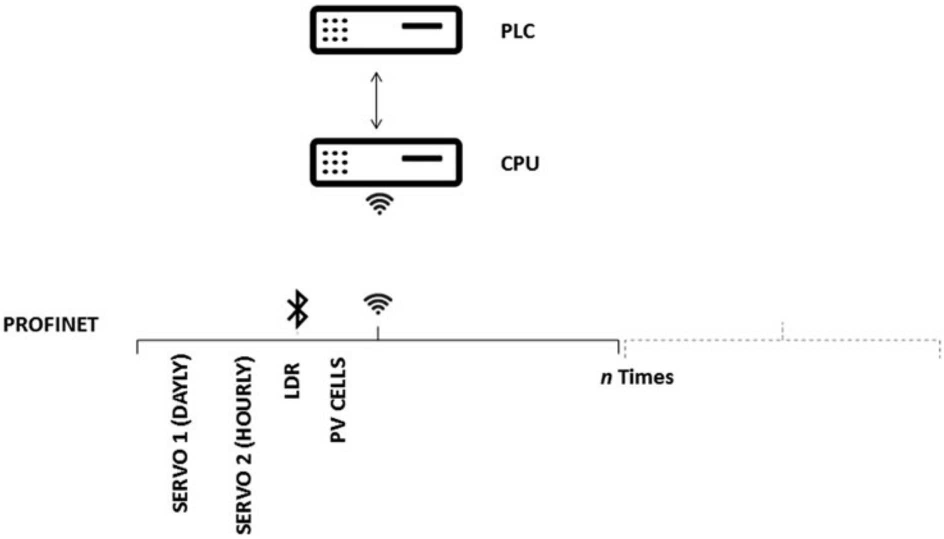

Fig. 26. Electronic diagram 2 (Author).

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

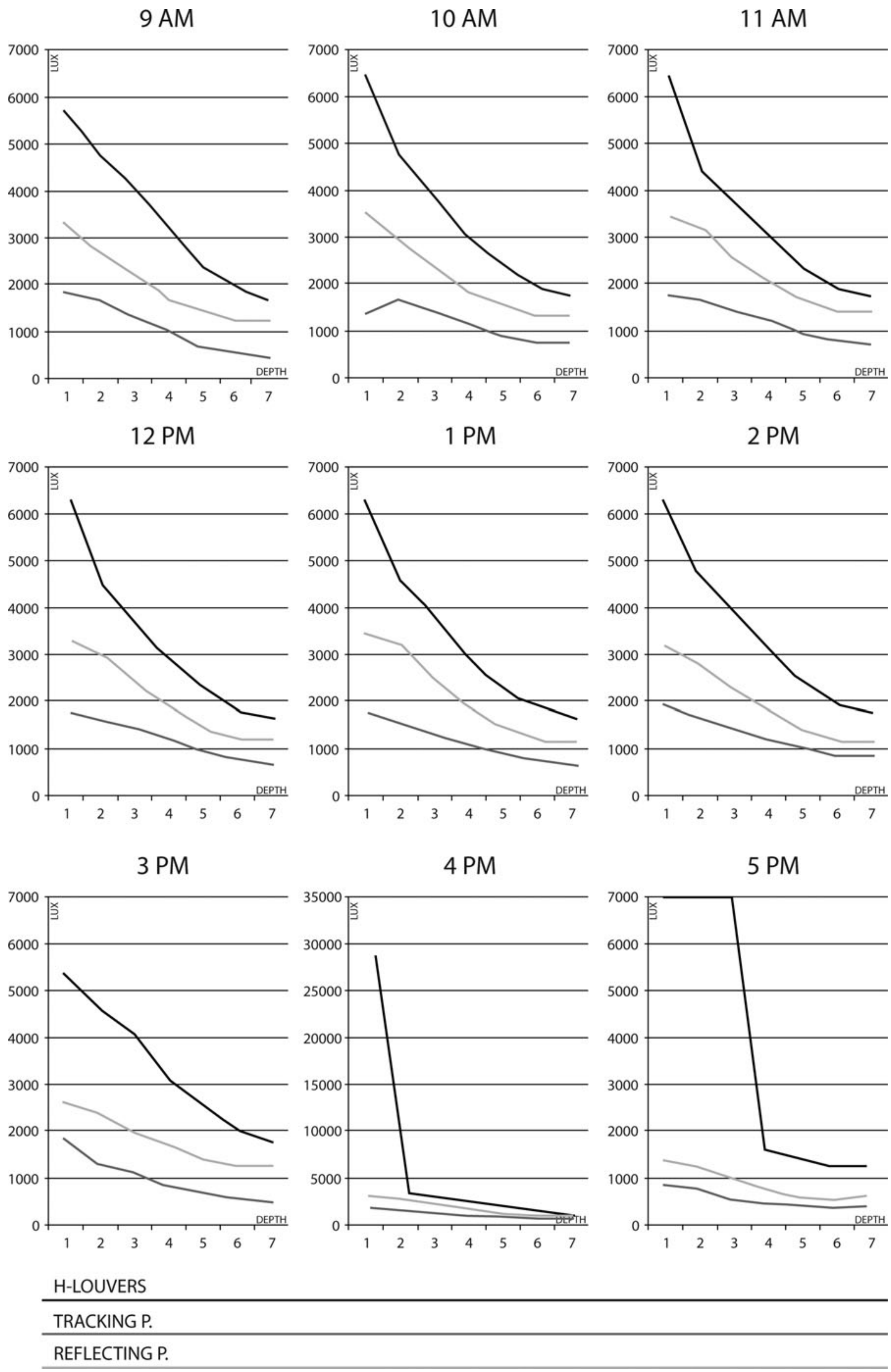

Fig. 27. Comparative results from simulations using three standard louvers against the system at tracking and reflecting states.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

more light taken at a threshold of 2000 lux, corresponding to tracking and reflecting states, respectively, according to the fuzzy logic implementation.

The graphs in Figure 27 show the results for light penetration in the described digital testing rig, where the vertical scale represents the lighting values in lux, and the horizontal the depth of the room, showing seven positions (each one placing a sensor or evaluation grid point) separated by 700 and 50 mm at each end for a total depth of 5 m.

The results provided in Figure 27 represent three different cases. The ones on the left show the results for a conventional horizontal louver system; the ones in the middle show results for a <2000 lux requirement (equivalent to tracking state), and the ones on the right show results for a >2000 lux requirement (equivalent to reflecting state). The results make evident a more uniform light distribution through the space for both reflecting and tracking positions compared to those of a conventional horizontal lover system, while simultaneously providing average levels close to the levels required by the user.

6. CONCLUSIONS AND FUTURE WORK

This paper presented a case study of design exploration supported by the implementation of computational processes at two levels, designer and user, where the problem of light penetration into an architectural space was explored and answered by a dynamic facade system. Digital daylight simulations demonstrated the high performance of the solution compared to a conventional horizontal louver system in terms of light modulation and distribution inside the covered space. Nonetheless, detailed analysis of such experimentations is left behind the scope of the present paper, particularly experimentation at the user level, which would require the physical implementation of the system at a 1:1 scale within a real space and time situation to record the responses and behavior of real users. It should be also noted that the final implementation was carried out for a single panel (whose position was replicated for daylight simulations), leaving uncovered the logic definition of the group. In this respect, the adoption and insertion of group-based behavior logics like cellular automata (Zawidzki, 2008) may offer interesting possibilities for further development.

Taking into account the current relevance of the subject due to the requirement for high-performance facade systems to help cope with sustainability requirements and made possible by the recent development and popularization of computational technologies, given its relative simplicity and functional adequacy, the dynamic sun-shading system here presented intends to be a contribution to the particular field of research. However, from a more general perspective, it is also clear that daylight and visual functioning are still limited areas of facade performance, and should be considered in tandem with other requirements, namely, thermal, acoustic, and aesthetical, of which at least the first two may be possible to integrate within a more complex logic to reach possibilities for general optimization of the system(s). Much simpler and more energy-efficient integrated systems could be developed (even if founded on very complex internal logics), that is, leaving complexity to integrated control processes in a virtual realm while keeping simple robust solutions at on a macroscale material level.

ACKNOWLEDGMENTS

The authors thank Mostapha Roudsari, Chris Mackey, and the entire developing team of Ladybug+Honeybee (http://www.grasshopper3d.com/group/ladybug) for creating tools that have provided important assistance to the work in this article.

REFERENCES

Dubois, M.-C. (2013). Visual protection devices for architectural applications: key issues and characteristics. *Proc. 8th Energy Forum Conf.: Advanced BuildingSkins*, pp. 135–140, Bolzano, Italy, November 5–6.

El Sheik, M. (2011). *Intelligent building skins: parametric-based algorithm for kinetic facades design and daylighting performance integration*. PhD Thesis. University of Southern California, Faculty of the USC School of Architecture. Accessed at http://digitallibrary.usc.edu

Fox, M., & Kemp, M. (2009). *Interactive Architecture*. Princeton, NJ: Princeton Architectural Press.

Gomide, F., Gudwin, R., & Tanscheit, R. (1995). Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. *Proc. 6th IFSA Congr. Tutorials*, São Paulo, Brazil, July.

Lee, C. (1990). Fuzzy logic in control system: fuzzy logic controller, part I and II. *IEEE Transactions on Systems, Man and Cybernetics 20*, 404–435.

Loomen, R. (2010). *Climate adaptive building shells: what can we simulate?* MS Thesis. Technische Universiteit Eindhoven.

Moloney, J. (2007). A framework for the design of kinetic façades. *Proc. CAAD Futures'07* (Dong, A., Vande Moere, A., & Gero, J.S., Eds.), pp. 461–474. New York: Springer.

Rossi, D., Nagy, Z., & Schlueter, A. (2012). Adaptive distributed robotics for environmental performance, occupant comfort and architectural expression. *International Journal of Architectural Computing 10*, 341–360.

Stevenson, C. (2011). Morphological principles of kinetic architectural structures. *Proc. Adaptive Architecture Conf.*, pp. 1–12, London, March 3–5.

Tanscheit, R. (1999). Sistemas suzzy. *Proc. DEE-PUC-Rio*, pp. 2–7, Rio de Janeiro, January.

Thün, G.M., Velikov, K., O'Malley, M., & Sauvé, L. (2012). The agency of responsive envelopes: interaction, politics and interconnected systems. *International Journal of Architectural Computing 10(3)*, 377–400.

Thün, G., & Velikov, K. (2013). Responsive envelopes: bridging environmental response and human interaction. *Proc. 8th Energy Forum Conf.: Advanced Building Skins*, pp. 317–321, Bolzano, Italy, November 5–6, 2013.

Urquiza, R. (2010). Parametric performative systems: designing a bioclimatic responsive skin. *International Journal of Architectural Computing 8(3)*, 279–300.

Velasco, R., Brakke, A.P., & Chavarro, D. (2015). Dynamic facades and computation: towards an inclusive categorization of high performance kinetic façade systems in computer-aided architectural design futures. The next city—new technologies and the future of the built environment. *Proc. 16th Int. Conf., CAAD Futures 2015* (Celani, G., Sperling, D.M., & Santos Franco, J.M., Eds.), pp. 172–191. Berlin: Springer.

Vollen, J.O., & Winn, K. (2013). Climate camouflage: advection based adaptive building envelopes. *Proc. 8th Energy Forum Conf.: Advanced Building Skins*, pp. 305–310, Bolzano, Italy, November 5–6.

Zadeh, L. (2008). Is there a need for fuzzy logic? *Information Sciences 178(13)*, 2751–2779.

Zawidzki, M. (2008). Implementation of cellular automata for dynamic shading of building façade. *Proc. ACADIA 2008 Conf.*, pp. 246–255. Minneapolis, MN, October 13–19.

Rodrigo Velasco is an Architect and is a Researcher at the Universidad Piloto de Colombia. He has an MEng in computa-

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press

tional design and has devoted more than 10 years to the study of building envelopes. His career includes research activities at the University of Nottingham, the University of Hong Kong, and the University Nacional de Colombia; teaching at other universities; and a variety of publications on the subject. He is a founding member and current Director at FRONTis3D, a company specializing in consultancy, fabrication, and fitting of nonstandard brise-soleil systems for buildings.

Rubén D. Hernández is a Mechanical Engineer and is currently Research Coordinator for the Mechatronics Engineering Programme at Universidad Piloto de Colombia. He has an MEng and is a PhD candidate in mechanical engineering.

Nicolás Marrugo is a Mechanical Engineer at Universidad Piloto de Colombia, where he worked as a Research Assistant on the present project.

César Díaz is an Architect at Frontis 3D, where he worked as Research Assistant on the present project.

https://doi.org/10.1017/S0890060415000463 Published online by Cambridge University Press